UNITED STATES PATENT APPLICATION

For

METHOD FOR FAST LABEL SWITCHOVER WITH MULTIPROTOCOL LABEL SWITCHING

Inventors:

Hariprasad Ginjpalli, Durai Chinnaiah,
Jayakumar Jayakumar

Prepared by:

BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026
(408) 720-8300

Attorney's Docket No.: 81862P265

"Express Mail" mailing label number: EL617182944US
Date of Deposit: September 27, 2001
I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service on the date indicated above and that this paper or fee has been addressed to the Assistant Commissioner for Patents, Washington, D. C. 20231

Leslie Rogan
(Typed or printed name of person mailing paper or fee)

Leslie D. Rogan
(Signature of person mailing paper or fee)

9/27/01
(Date signed)

METHOD FOR FAST LABEL SWITCHOVER WITH MULTIPROTOCOL LABEL SWITCHING

FIELD OF THE INVENTION

[0001] The present invention relates generally to the field of network communication. More specifically, the present invention relates to a method and a system for providing fast label switchover for layer 2 transport over Multiprotocol Label Switching (MPLS).

BACKGROUND

[0002] Multiprotocol Label Switching (MPLS) is a technology for backbone networks. **Figure 1** is a diagram illustrating an example of a prior art MPLS network with edge routers and switch routers R0 to R6. Label Edge Routers (LERs) 100, 125 sit at edges of the MPLS network. The LER 100 is capable of encapsulating an MPLS label 103 to the datagram 102. In addition to the LERs 100 and 125, there may be multiple Label Switch Routers (LSRs) 105-120 responsible for forwarding the datagram 102. The LSR 105 also encapsulates the datagram 102 with another label 104. The LSRs 105-120 communicate with each other using a Label Distribution Protocol (LDP). The LER 100 is referred to as ingress LER, and the LER 125 is referred to as egress LER.

[0003] The datagram 102 travels from the LSR 105 to the LSR 120 through the MPLS network using a Label Switched Path (LSP) 135. The LSR 105 forwards the datagram 102 to its adjacent LSR 110. The LSR 110 then forwards the datagram 102 to its adjacent LSR 115, and so on. MPLS uses a hierarchy of labels known as label stack. For layer 2 transport over MPLS applications, the

label stack includes a virtual circuit (VC) label and one or more tunnel labels above the VC label.

[0004] Referring to **Figure 1**, the LER 100 pushes the first label 103 onto the label stack. This is the VC label. The VC label is determined based on information associated with the datagram 102. Thus, the VC label identifies a virtual circuit in the egress LER 125 that the ingress LER 100 wants the datagram 102 is to be forwarded to. Each VC label is associated with a VC.

[0005] The LSR 105 pushes the second label 104 on the label stack. This is the tunnel label. The tunnel label identifies the adjacent LSR 110 that the datagram 102 is to be forwarded to. This continues until the datagram 102 is forwarded to the LSR 120. The LSR 120 removes the current tunnel label from the top of the label stack but does not replace that tunnel label with another tunnel label because the operation associated with the tunnel label is a pop operation. Through the tunnel label, the LSR 120 also recognizes its adjacent router, which is the LER 125. When the LER 100 is adjacent to the LER 125, the VC label 103 is the only label pushed onto the label stack. Thus, the VC label is normally on the label stack, and the tunnel label, if present, is immediately above the VC label.

[0006] When a failure prevents the datagram 102 from being forwarded to the egress LER 125 along the LSP 135, the tunnel labels in the label stack may need to be modified. It is essential that this process be carried out quickly to minimize delay.

SUMMARY OF THE INVENTION

[0007] A method of rewriting tunnel labels in a multiprotocol label-switching (MPLS) network is disclosed. For one embodiment, one or more virtual circuit (VC) labels associated with a tunnel label are mapped to a group identification (GID). The tunnel label is mapped to the GID in a GID table. When the one or more VC labels are to be associated with a new tunnel label, the tunnel label in the GID table is changed to the new tunnel label.

[0008] Other features and advantages of the present invention will be apparent from the accompanying drawings and from the detailed description which follows.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The present invention is illustrated by way of example, and not limitation, in the figures of the accompanying drawings in which like references indicate similar elements and in which:

[0010] **Figure 1** is a diagram illustrating an example of a prior art MPLS network with edge routers and switch routers.

[0011] **Figure 2A** is a diagram illustrating an example of a prior art MPLS network with multiple Label switch paths (LSPs) and multiple virtual circuits (VC).

[0012] **Figure 2B** is a diagram illustrating an example of a backup LSP in an MPLS network.

[0013] **Figure 3** illustrates a prior art label table used in transporting asynchronous transfer mode (ATM) traffic over MPLS.

[0014] **Figure 4** is a label table that illustrates one embodiment of mapping VC labels to group identifications (GIDs).

[0015] **Figure 5** is a GID table that illustrates one embodiment of mapping GIDs to tunnel labels.

[0016] **Figure 6A** is a diagram illustrating a datagram with a VC label and an existing tunnel label.

[0017] **Figure 6B** is a diagram illustrating the same datagram as in **Figure 6A** with a new tunnel label replacing the existing tunnel label.

[0018] **Figure 7** is a diagram illustrating an example of a link reroute in an MPLS network.

[0019] **Figure 8** illustrates another embodiment of the GID table.

[0020] **Figure 9** illustrates enabling of backup labels using the GID table.

[0021] **Figure 10A** illustrates a datagram with a VC label and an existing tunnel label.

[0022] **Figure 10B** illustrates the same datagram as in **Figure 10A** with an addition of a backup tunnel label

[0023] **Figure 11** illustrates a computer system that can be used with one embodiment of the present invention.

DETAILED DESCRIPTION

[0024] For one embodiment, a method and a system for rewriting tunnel labels in a multiprotocol label switching (MPLS) network is disclosed. Each of one or more tunnel labels supported by an MPLS router is mapped to a different group identification (GID) in a GID table. In a label table, each of one or more virtual circuit (VC) labels supported by the MPLS router is mapped to a GID corresponding to the tunnel label that the VC label is associated with. When one or more different tunnel labels are to be rewritten, the one or more different tunnel labels are rewritten in the GID table.

[0025] **Figure 2A** is a diagram illustrating an example of a prior art MPLS network with multiple Label switch paths (LSPs) and multiple virtual circuits (VC). Datagram 202 is to be forwarded from the ingress LER 100 to an egress LER 225 using LSP 235. A VC label 203 is pushed on top of the label stack associated with the datagram 202. The VC label 203 indicates that the datagram 202 is to be forwarded to the egress LER 225. The LSR 105 pushes a tunnel label 204 above the VC label 203 to indicate that the datagram 202 is to be forwarded to the LSR 110.

[0026] The tunnel labels 104 and 204 are not be the same because the LSPs 135 and 235 are different. Furthermore, because the LSR 110 forwards the datagram 102 to the LSR 115 and forwards the datagram 202 to the LSR 130, the tunnel label 106 is different from the tunnel label 206. Each LSR stores the VC label information and the tunnel label information in a database referred to as a label information base (LIB). For example, the LIB in the LSR 110 includes

information that maps the tunnel label 104 to the tunnel label 106 and information that maps the tunnel label 204 to the tunnel label 206. The LSRs are not even aware that a VC label exists below a tunnel label. From the standpoint of the LSR 110, the tunnel labels 104 and 204 are incoming tunnel labels, and the tunnel labels 106 and 206 are outgoing tunnel labels.

[0027] The link between two adjacent LSRs is also referred to as a tunnel LSP. A single tunnel LSP is capable of carrying thousands of different VCs. As the transport of layer 2 data over MPLS becomes more popular, the number of VCs that get mapped to a single tunnel LSP increases. When a failure occurs (e.g., link failure, node failure, etc.) and an alternate tunnel LSP is used, the LIB has to be updated. For example, when the link between the LSR 105 and the LSR 110 fails, the LIB of the LSR 105 has to be updated. Typically, this involves rewriting the tunnel labels for all of the VCs that are previously mapped to the outgoing tunnel labels 104 and 204.

[0028] An LSP (or path) is controlled by a head-end router. For example, the LSP 135 is controlled by the head-end LSR 105. The head-end LSR 105 monitors the status of the LSP 135 by receiving information from other LSRs 110, 115, and 120 along the LSP 135. When there is an error along the path, the head-end router receives error information from the appropriate router. For example, when the link between the LSR 110 and the LSR 115 is broken, the LSR 105 receives error information from the LSR 110 and concludes that the LSP 135 has failed. The head-end LSR 105 then determines a backup LSP. **Figure 2B** is a diagram illustrating an example of a backup LSP in an MPLS network. The backup LSP

240 is used to forward datagrams from the head-end LSR 105 when the primary LSP 135 fails due to a broken link between the LSR 110 and the LSR 115. The backup LSP 240 may be previously pre-configured or dynamically computed by the head-end router 105. The process of switching from the original LSP 135 to the backup LSP 240 is referred to as a path protection process and may require rewriting multiple tunnel labels.

[0029] **Figure 3** illustrates a prior art label table used in transporting asynchronous transfer mode (ATM) traffic over MPLS. The label table may be part of the LIB database stored in a LSR. The label table includes an ATM identification column 305, a VC label column 310, a tunnel label column 315, and an interface column 320. The ATM identification column 305 lists each virtual circuit (VC) identified by its VPI/VCI. VPI is a virtual path identifier that identifies a path that an ATM datagram (or cell) will take. VCI is a virtual channel identifier that identifies a channel within the path that the ATM datagram will take to reach its destination. There are four entries in this label table to identify four different VCs and VC labels supported by the LSR. Each of the four VC labels is associated with a tunnel label listed under the tunnel label column 315. Thus, when the LSR receives an ATM datagram with a VC label having a value of "100", the LSR adds an outgoing tunnel label having a value of "20" to that ATM datagram. The ATM datagram is then sent out to the interface "1" of the LSR. In this example, adding the outgoing tunnel label having the value of "20" is similar to pushing that tunnel label on top of the label stack, as previously described.

[0030] The label table in **Figure 3** indicates that there are three VC labels that are mapped to the outgoing tunnel label having the value of "20". These three VC labels have the value of "100", "200", and "400" respectively. The VC label having the value of "300" is mapped to an outgoing tunnel label having a value of "30". The label table lists an outgoing tunnel label for every VC label, even though the same outgoing tunnel label may be repeated multiple times.

[0031] When the LSR supports a large number of VCs, the label table in **Figure 3** can be very large. Typically in the prior art, when a path failure occurs that necessitate changing or rewriting an existing outgoing tunnel label to a new outgoing tunnel label, each corresponding entry in the label table has to be changed. For example, using the label table in **Figure 3**, when the existing outgoing tunnel label having the value of "20" is to be changed to a new outgoing tunnel label having a value of "40", three entries have to be changed. When there are thousands of VC labels mapped to the same existing outgoing tunnel label, thousands of outgoing tunnel labels in the label table have to be changed. In addition to changing the outgoing tunnel labels, the corresponding interfaces may also need to be changed. For example, the interface column 320 indicates that an interface is changed from "1" to "3" to correspond to the new outgoing tunnel label having the value of "40".

[0032] **Figure 4** is a label table that illustrates one embodiment of mapping of the VC label to group identifications (GIDs). The label table in **Figure 4** includes an ATM identification column 405, a VC label column 410 and a GID column 415. For one embodiment of the present invention, each VC label that was previously

mapped to the same tunnel label is mapped to one group identification. For example, the label table in **Figure 4** indicates that the VC labels having the values "100", "200", and "400" are mapped to the GID having a value of "1". Note that these same VC labels are mapped to the same tunnel label having the value of "20" in the label table in **Figure 3**. Because the VC label having the value of "300" is mapped to a different tunnel label, it is mapped to a different GID (having a value of "2").

[0033] **Figure 5** is a GID table that illustrates one embodiment of mapping GIDs to tunnel labels. The GID table illustrates mapping of the GID to the tunnel label. The GID table includes a GID column 520, a tunnel label column 525, and an interface column 530. Each GID listed in the GID column 520 is mapped to a different tunnel label listed in the tunnel label column 525. For example, the GID having a value of "1" is mapped to the tunnel label having the value of "20". The GID having a value of "2" is mapped to the tunnel label having the value of "30", etc. The GID table and **Figure 5** and the label table in **Figure 4** together provide the same VC label-to-tunnel label mapping information as the label table in **Figure 3**. However, by using the GID table, when a failure occurs that necessitate changing or rewriting the existing tunnel labels, the number of changes required is greatly reduced. For example, when a tunnel label mapped to the GID having the value of "1" needs to be changed, the corresponding entry is changed from an existing value of "20" to a new value of "40", and the corresponding interface is also changed from "1" to "3", as illustrated in the GID table. Note that the same change applies regardless of the number of VC labels mapped to the tunnel label being changed. **Figure 6A** is a diagram illustrating a datagram with a VC label

600 and an existing tunnel label 605. **Figure 6B** is a diagram illustrating the same datagram as in **Figure 6A** with a new tunnel label 610 replacing the existing tunnel label 605.

[0034] The technique of replacing an existing tunnel label with a new tunnel label using a GID table can be used to switch from a failed LSP to a backup LSP in the path protection process described above. That is, when a LSP fails, the tunnel labels associated with the failing LSP are replaced with new tunnel labels associated with the backup path. The GID table allows the rewriting of the tunnel labels to be performed quickly and efficiently.

[0035] A link is controlled by the end routers associated with that link. When a link fails, link reroute to a backup link can be performed. **Figure 7** is a diagram illustrating an example of a link reroute in an MPLS network. When the LSR 110 recognizes that the link between it and the LSR 115 fails, the LSR 110 may reroute the datagrams intended to be forwarded to the LSR 115 to the LSR 130. This is done using a backup link between the LSR 110 and the LSR 130. Note that this requires pushing a backup label 206 onto the label stack above the tunnel label 106. Except for the backup label 206, the format of the datagram to be transmitted between the LSR 110 and the LSR 115 remains the same when that datagram is forwarded from the LSR 110 to the LSR 130. Thus, the LSR 110 swaps the tunnel label 104 with the tunnel label 106 (as before) but also pushes the backup label 206 on the top of the label stack. The LSR 130 recognizes that the backup label is on top of the label stack, removes it from the label stack, and forwards the datagram to the LSR 115. The LSR 115 receives the identical

datagram from the LSR 130 as it would receive from the LSR 110. The process of rerouting from one link to a backup link is referred to as a link protection process. Typically, link protection is configured on a per link basis and the backup link may be preconfigured.

[0036] **Figure 8** illustrates another embodiment of the GID table. In addition to the GID column 520, the tunnel label column 525, and the interface column 530, this GID table also includes a backup label enable column 835, a backup label 840, and a new interface column 845. When an entry in the backup label enable column 835 is listed as "NO", the corresponding entries listed in the backup label column 840 and the new interface column 845 are ignored. In this example, because all of the entries under the backup label enable column 835 are listed as "NO", the GID table in **Figure 8** is similar to the GID table in **Figure 5**.

[0037] For one embodiment, when an entry under the backup label enable column 835 is listed as "YES", the GID table in **Figure 8** can be used to add a backup label in a link protection process. **Figure 9** illustrates enabling of backup labels using the GID table. The first entry in the backup label enable column 835 is listed as "YES". This indicates that a backup tunnel label having a value of "50" is pushed onto the label stack, as shown under the backup label column 840. Note that the corresponding tunnel label shown under the tunnel label column 525 and the corresponding interface shown under the original interface column 530 remain unchanged at "20" and "1", respectively. The rerouted datagram is forwarded to a new LSR using the interface shown under the new interface column 845.

[0038] **Figure 10A** illustrates a datagram with a VC label 1000 and a tunnel label 1005. **Figure 10B** illustrates the same datagram being rerouted with the same VC label 1000, tunnel label 1005, and a backup label 1010.

[0039] **Figure 11** illustrates a computer system that can be used with one embodiment of the present invention. The operations of the various methods of the present invention may be implemented by a processing unit 1105 in a digital processing system 1100. The processing unit 1105 is coupled with a bus 1110 and executes sequences of computer program instructions 1115. For example, the computer program instructions 1115 may include instructions to perform the mapping of tunnel labels to GIDs in a GID table and the rewriting of the tunnel labels in the GID table. The instructions 1115 may be stored in a memory which may be considered to be a machine-readable storage media 1120. The machine-readable storage media 1120 may be used with a drive unit 1125 coupled with the bus 1110. The instructions 1115 may be loaded into main memory 1130 coupled with the bus 1110. Although not shown, other memory devices such as, for example, read-only memory, a persistent storage memory, or any combination of these devices may also be used to store the instructions 1115.

[0040] Execution of the sequences of instruction 1115 causes the processing unit 1105 to perform operations according to the present invention. The instructions 1115 may be loaded into the main memory 1130 from a storage device or from one or more other digital processing systems (e.g. a server computer system) over a network 1135 using network interface device 1140. In

other cases, the instructions 1115 may not be performed directly or they may not be directly executable by the processing unit 1105. Under these circumstances, the executions may be executed by causing the processing unit 1105 to execute an interpreter that interprets the instructions 1115, or by causing the processing unit 1105 to execute instructions, which convert the received instructions 1115 to instructions which can be directly executed by the processor. Other devices may also be connected to the bus 1100 including, for example, a video display 1145, an alpha-numeric input device 1150, and a cursor control device 1155.

[0041] In other embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to implement the present invention. Thus, the present invention is not limited to any specific combination of hardware circuitry and software, nor to any particular source for the instructions executed by the digital processing system 1100.

[0042] A method and a system for fast label switching for layer 2 transport over MPLS networks have been disclosed. The method minimizes the switchover time for path protection and link protection. The method also allows for memory conservation because the number of tunnel labels stored in the label information base (LIB) is significantly less than the number of VC labels. Although the above description refers to path failure and link failure, the method and system disclosed may also be applied when there is a node failure.

[0043] Although the present invention has been described with reference to

specific exemplary embodiments, it will be evident that various modifications and changes may be made to these embodiments without departing from the broader spirit and scope of the invention as set forth in the claims. Accordingly, the specification and drawings are to be regarded in an illustrative rather than a restrictive sense.